Exogenous Update Scheduling in the Industrial Internet of Things for Minimal Age of Information

Leonardo Badia, *Senior Member, IEEE*, and Andrea Munari, *Senior Member, IEEE*

***Abstract*—Data freshness is extremely important for real-time applications and generally measured with age of information (AoI). Related studies typically assume that fresh data can be generated at any time. However, in industrial Internet of Things (IIoT) applications, such as alerting, monitoring, or task-oriented operations, data generation is often exogenous and occurs within a finite window. This motivates our analysis, where we investigate AoI-minimizing scheduling for status updates from an IIoT source that generates fresh data only at random intervals. We differentiate between infinite and finite horizons, with the latter being more aligned with IIoT tasks. For each scenario, we examine both agnostic (predefined and unchangeable) and source-aware scheduling, based on the probability of fresh data generation and the duty cycle. We provide a tight bound for source-aware scheduling in the infinite-horizon case and exact expressions for the other scenarios. We assess the increase in AoI from sporadic data generation, finding worst-case factors of 3 for agnostic scheduling and 2 for source-aware scheduling. However, these estimates are pessimistic when the data generation probability is at least an order of magnitude higher than the duty cycle. In such cases, the AoI increase is less than 20% for agnostic scheduling and almost negligible for source-aware scheduling.**

***Index Terms*—Age of information (AoI), industrial Internet of things (IIoT), scheduling, sensor networks.**

I. Introduction

AGE of Information (AoI) is a performance metric quantifying the freshness of status updates. When considering a remote source transmitting data to a receiver, AoI at a given instant is defined as the time elapsed since the generation of the most recently received message [1]. Thanks to its analytical character, AoI has been the subject of a flourishing line of research that combines system modeling, distributed control, and performance evaluation.

AoI is commonly analyzed under the assumption that data are *generated at will*, i.e., a new message with fresh information is always available at the source [2], [3]. However, many industrial scenarios are event-driven and follow a different rationale. For example, alert systems generate data only in response to certain triggers, rather than as constant streams of data [4]. Production monitoring systems also follow specific events in the manufacturing process [5]. Correlated generation is also possible [6], in which case data can be inferred from a set of multiple sensors, not all controllable. Finally, data can be generated after a sensor reading or a user query [7], meaning that there is no further fresh update to send in between these inputs.

We investigate the minimization of the average AoI for a system consisting of three components: 1) an exogenous source that generates updates about a process of interest; 2) a transmitter that can send them to 3) a receiver, where AoI is computed, with a limitation in the transmission cycle, consistent with scenarios of the industrial Internet of things (IIoT) [8], [9], [10]. For instance, commercial technologies like LoRaWAN limit the operational duty cycle (e.g., to 1% for the ISM band) due to technological and regulatory constraints.

We consider both an infinite and a finite-time horizon, the former being commonly adopted in most literature [11], [12], [13], [14]. Yet, the latter corresponds to a more realistic setup where the monitoring task has a predefined duration [15], [16], [17]. In parallel to finite/infinite horizon, we investigate two different kinds of scheduling: 1) source-agnostic, where transmissions are allotted at predetermined time instants based on a prior probability of data availability, and 2) source-aware adaptive scheduling. [18]. The latter follows from a memory-less control policy obtained through dynamic programming, consequent to framing the problem as a Markov decision process (MDP) [19]. We overall present four different solutions (finite/infinite horizon and agnostic/aware scheduling).

The purpose of the analysis is to quantify the increase in AoI due to exogenous updates. We formally prove, as theoretical upper bounds, that exogenous updates can cause the AoI to *double* when using a source-aware scheduler, and *triple* for an agnostic scheduler. This occurs when the frequency of update generation is low (i.e., approaching the transmission duty cycle). As the update generation rate increases, the surge in AoI is much lower, especially for a finite time horizon.

We also provide quantitative evaluations concerning the variability of a source-aware scheduler, the role of the time horizon, and the comparison of agnostic versus aware scheduling. These

Received 28 February 2024; revised 7 June 2024, 9 August 2024, and 8 September 2024; accepted 25 September 2024. This work was supported in part by the European Union under the Italian National Recovery and Resilience Plan (NRRP) of NextGenerationEU, partnership on “Telecommunications of the Future” (PE00000001 - program “RESTART”), and in part by the Federal Ministry of Education and Research of Germany in the programme ‘Souverän. Digital. Vernetzt.’ Joint project 6G-RIC, project identification number 16KISK022. Paper no. TII-24-0896. *(Corresponding author: Leonardo Badia.)*

Leonardo Badia is with the Department of Information Engineering, University of Padova, 35131 Padova, Italy (e-mail: leonardo.badia@unipd.it).

Andrea Munari is with the Institute of Communications and Navigation, German Aerospace Center (DLR), 82234 Weßling, Germany (e-mail: andrea.munari@dlr.de).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TII.2024.3475422.

Digital Object Identifier 10.1109/TII.2024.3475422

evaluations are often missing in the literature, despite being interesting for practical implementations. For example, sensors with preprogrammed scheduling can be simpler to manage, whereas an adaptive scheduler requires more sophisticated control and persistent data acquisition [18], [20]. Our analysis allows us to investigate whether the latter are worth implementing.

In summary, we give the following novel contributions. First, we present a model for exogenous update generation that makes it possible to relax the standard framework for AoI analysis in the literature, which usually assumes generate at will. Moreover, we apply this extension to all possible cases of finite/infinite horizon and source aware/agnostic scheduling, discussing how the optimal transmission scheduling is affected in all cases. For the case of infinite horizon and source-aware scheduling, we discuss how tight low-complexity approximations can be obtained from the solution of the finite horizon case. Finally, we evaluate this extension, for both theoretical upper bounds and practical cases where the surge in AoI is limited.

II. Related Work

Many papers investigate AoI in communication systems, especially in the context of remote sensing for the Internet of Things. Their approaches relate to various extents to ours, even though none of them directly addresses the specific topic of exogenous update generation. Actually, most of the literature dealing with goal-oriented communication [21], [22], [23] assumes the opposite, i.e., the sensing units can choose when to transmit data based on the state of the network.

AoI is usually addressed as a performance evaluation metric within queueing theory or medium access. For example, seminal papers [1] and [24] investigated various queueing disciplines and the role of carrier-sense, respectively. These two directions were further expanded in the literature to more queueing systems [2], [25] and access protocols [15], [26].

Other studies explore AoI in the context of network optimization, as a guiding parameter for a closed-loop network control. For example, Fountoulakis et al. [27] considered a scheduling of updates with AoI constraints, under the standard objective of throughput optimization. Similarly, Li et al. [12] considered a cyclic scheduler for an infinite time horizon, with an AoI threshold. The effect of duty cycle limitations is considered in [10], but the underlying medium access is based on an ALOHA protocol with an AoI threshold, and the horizon is taken as infinite.

Xie et al. [5] and Ceran et al. [11] studied AoI under automatic repeat request (ARQ), where the decision on how and when to perform retransmissions can be compared to our problem, with the differences that we consider multiple packets instead of one at a time, and do not apply retransmissions. In addition, Xie et al. [5] can be seen as an agnostic scheduler subject to a zero wait generation policy [4], whereas Ceran et al. [11] was akin to a source-aware scheduler with infinite horizon. A similar idea is exploited in [9], where fresh data are always available at the source, and ML is applied to estimate the probability of missed delivery to adjust the scheduling.

In a similar spirit, Javani et al.[28] analyzed status updates under erasures and with feedback. The authors do not consider exogenous generation and study an infinite horizon, over which they adjust the schedule to account for erasures. Their main constraint is not a duty cycle limitation, but the updates are assumed as consisting of multistructured data, so that partial reception is possible. The ultimate choice is whether to keep a partially delivered update, which is older, or discard it in favor of a new one. A similar problem, albeit to a more limited extent (only one retransmission), is considered in [29].

Xie et al. [8], Liu et al. [13], Kuo [15], and Liu et al. [16] considered scheduling in the sense of source selection among a pool of candidates, as opposed to choosing the temporal placement of transmissions, as we do here. While these proposals share similarities with the present paper, they are ultimately different. For example, Liu et al. [13] considered a source-aware scheduling for an infinite horizon with an AoI guarantee. In addition to source selection, multihop routing (seen as a scheduling of links) so they have constraints in the number of transmissions, but also related to ingress and egress of nodes and mutual interference.

In [18], the reverse scheduling process (one-to-many) is considered, where a single source feeds data to multiple applications and must do so in a timely manner, prompting the authors to propose an AoI-aware strategy rather than a periodic, agnostic scheduler. A time division scheduling is considered in [15] to avoid interference, and in [16], the problem is translated from a time scheduling to the choice of a sequence pattern avoiding collisions and lowering AoI. Xie et al. [8] discussed a guided exploration Q-learning procedure to decompose the scheduling layers. A similar approach is adopted in [30], with a learning procedure, but also allowing for coding multiple information content.

Finally, Akar and Gamgam [31] and Lin et al. [32] assumed a three-stage network similar to ours, where data collection is split into separate sources and a transmission unit. However, their focus is on source selection for minimal AoI, considering the varying urgency of different sources, all of which always have fresh content. In contrast, we consider a single source that may not always have new information to deliver.

Another line of research combines AoI scheduling with energy harvesting [33], where the schedule accounts for erratic energy availability. This is similar to our problem, although ours pertains to data unavailability instead. This concept is expanded in [34] to address AoI-minimizing stateful scheduling over a variable channel with a power constraint and an infinite time horizon. Conversely, Hatami et al.[14] incorporated the battery level into the system state to make the decision of whether to transmit, while Gindullina et al. [35] extended the approach to scenarios with multiple energy sources.

Last, Sun et al. [4] assumed that the "generate at will" assumption does not hold, although the transmitter can control update generation. They introduce the concept of zero-wait policy and present notable results, such as this policy not being optimal if immediate transmission of updates leads to insufficient novelty at the receiver's side. Our approach differs in that the novelty of updates depends on an exogenous process, over which the transmitter has no control and may not even be informed about in the case of an agnostic scheduler.

III. System Model

Consider the system represented in Fig. 1, consisting of: (a) a source of information monitoring a remote process; (b) a

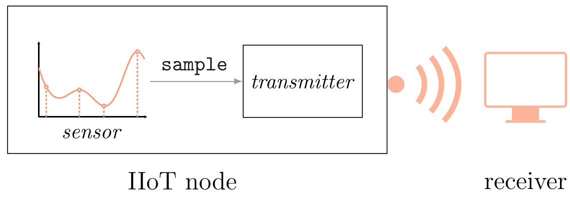


Fig. 1. System components.

transmitter, directly related to the source, but logically distinct; (c) a receiver, i.e., the end point of the system. While an IIoT node used for *data collection* may combine the first two elements [8], [9], the key point of our investigation is that the source may generate data only sporadically, to which the transmitter ought to adjust. The transmitter–receiver pair represents a *delivery network* sometimes studied as a queueing system. We take the perspective of studies like [28], where the processing time at the receiver's is immediate (or, it is equal to one slot). Since the arrivals per slot are never more than one packet, there is no queueing at the receiver's buffer. The transmitter has a one-packet buffer with preemption, i.e., only considers the most recent packet generated [25]. Finally, similar to these references, we neglect the propagation delay between the transmitter and receiver, as it would introduce only a constant bias. All these assumptions are noncritical and can be relaxed by incorporating a queueing analysis. The key point is that, in scheduling the updates, the transmitter should account for the availability of fresh information generated by the source, over which it has no direct control.

We assume that once updates are scheduled, their transmission is always successful. If channel impairments or collisions lead to the loss of update packets, this would occur only *after* the transmission has been scheduled. Therefore, the optimality of the policy under these conditions can only be assessed in expectation. Nonetheless, this issue has been investigated in other papers [20], [28], and can be considered an immediate extension of the work presented here.

We take a discrete time axis divided into slots, each corresponding to the time for the transmission of an update. The discrete time index $n \in \mathbb{N}$ can be seen as the sampling of a continuous time $t \in \mathbb{R}$ at equally spaced intervals of duration T. For quick reference, the nth time slot represents the interval $((n-1)T, nT]$, i.e., everything happening in between two sampling instants is observed in the latter of the two. We capture generation of fresh updates through a binary indicator value $\mathcal{G}[\cdot]$ whose meaning is that $\mathcal{G}[n] = 1$ if a status update is generated and made available to the transmitter during the nth time slot, i.e., at a certain time t between $(n-1)T$ and nT, and $\mathcal{G}[n] = 0$ otherwise. The transmission of updates is similarly represented through a binary indicator $\mathcal{T}[\cdot]$, with $\mathcal{T}[n] = 1$ if a transmission happens in slot n, $\mathcal{T}[n] = 0$ otherwise. We place both generations and transmissions at the end of the time slots, with the generation happening just before the transmission. Thus, $\mathcal{G}[n] = \mathcal{T}[n] = 1$ means that an update is generated *and immediately transmitted* in the same slot n.

One way to include the *generate at will* assumption is to imply that $\mathcal{G}[n] = 1$ for every n. Alternatively, one can take the source and the transmitter as having the same controller, so that a fresh status update is always available whenever needed, i.e., $\mathcal{G}[n] \geq \mathcal{T}[n]$. Instead, we consider *exogenous* information, meaning the transmitter cannot force a new measurement. Fresh data may or may not be generated in each time slot and the transmitter can wait before transmitting them.

A. AoI and Scheduling Policies

AoI in time slot n, denoted as $\delta[n]$, is defined as the difference between n and the value $\sigma[n]$ representing the last generation time of an update that was also transmitted, i.e.,

$$\delta[n] = n - \sigma[n] = n - \max_{k \leq z[n]} \{k : \mathcal{G}[k] = 1\}$$
$$\text{where: } z[n] = \max_{k \leq n} \{k : \mathcal{T}[k] = 1\} \ . \tag{1}$$

According to this definition, an update generated in a given time slot and transmitted immediately resets AoI to 0 at the receiver's side. This assumes negligible transmission and propagation delay, which can be relaxed by extending the analysis along the lines of [36] with more complex computations.[1]

To represent the generation of updates, we assume that they happen with independent identically distributed (i.i.d) probability ω in each slot. In other words, arrivals of fresh updates are Bernoulli with parameter ω, i.e.,

$$\mathbb{P}[\mathcal{G}[n] = k] = \omega k + (1-\omega)(1-k)$$
$$\text{for } k \in \{0, 1\}, \text{ for all} n. \tag{2}$$

The choice of Bernoulli arrival does not cause any loss of generality as the analysis of this article can be promptly extended to different generation processes [8], [37].

If two transmissions are scheduled in time slots n_1 and $n_2 > n_1$, the latter may have fresher information available. Thus, the second transmission decreases the AoI, albeit not necessarily to 0, but rather to the difference $n_2 - \sigma[n_2]$, where $\sigma[n_2] = \max\{\tau : \mathcal{G}[\tau] = 1, \tau \leq n_2\}$ is the instant of the last generation before n_2, see (1). In particular, if $\mathcal{G}[n_2] = 1$ then the transmission in n_2 resets AoI to 0. Conversely, if no update was generated in between, which happens with probability

$$p_m(n_2 - n_1) = (1-\omega)^{n_2 - n_1} \tag{3}$$

then $\sigma[n_1] = \sigma[n_2]$ and the transmission in slot n_2 carries the same update already delivered at time n_1. Fig. 2 reports an example of AoI evolution.

Updates that do not lower AoI ought to be avoided, to save transmissions. However, if the scheduling is planned in advance, such flexibility is not possible. We distinguish between *source-aware* and *source-agnostic* scheduling to describe the option (or lack thereof) for altering the plan as the scheduling unfolds, based on the current AoI and/or the availability of fresh updates. Over a finite time horizon of length N, our objective is to minimize the expected average AoI

$$\Delta = \mathbb{E}\left[\frac{1}{N}\sum_{n=1}^{N} \delta[n]\right] \tag{4}$$

where the expectation is computed on all the realizations of the update generation patterns. We remark that Δ involves both a

[1]The literature is divided on whether the AoI should reset to 0, as in [20], [28], or to 1, as in [1], [12], after a fresh update. This is merely a convention and does not affect the results except for a constant shift.

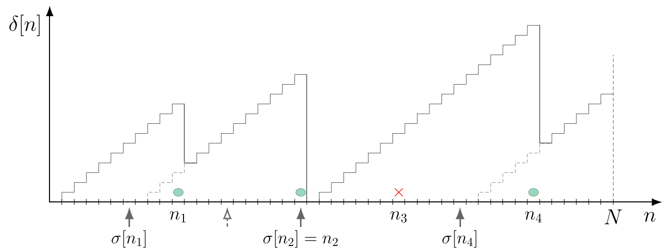


Fig. 2. Example of AoI evolution over time. Updates are generated in slots with an arrow, and transmissions happen in slots n_1, n_2, n_3, n_4. AoI is reset to 0 only in n_2 (also a generation instant). The "empty" arrow is an update not exploited, as the one in n_2 replaces it. For n_1 and n_4, AoI is reset to updates in $\sigma[n_1]$ and $\sigma[n_4]$, respectively. AoI is not reset in n_3 (marked with $\times$) due to the lack of updates since n_2.

TABLE I
NOTATION ADOPTED IN THE ARTICLE

symbol	meaning
$\mathcal{T}[n]$	1 if transmitting in slot n, 0 otherwise
$\mathcal{G}[n]$	1 if update generation in slot n, 0 otherwise
$\delta[n]$	instantaneous AoI in slot n
ω	per-slot probability of generating a fresh update
N	no. of slots in a time window
M	limit to the transmission opportunities in a time window
$d = M/N$	maximum average duty cycle allowed
$C = N/M$	average inter-transmission duration
Δ	expected average AoI

time average and a statistical expectation, yet we refer to this scenario simply as *average AoI* minimization [1], [17], [28].

We take an initial condition $\delta[0] = 0$, and we constrain the number of transmissions that can happen within the N slots. For numerical convenience, in the finite time horizon we set this to $M - 1$, the reason being that this corresponds to an upper limit on the *duty cycle* of the transmitter activity, defined as $d = M/N$, so that the average inter-transmission duration is set to $C = N/M$. In principle, the duty cycle can be less than or equal to M/N, but it is immediate that the equality must hold. Setting the maximum number of transmissions to $M - 1$ follows from implicitly equalling the initial condition $\delta[0] = 0$ to another transmission that is outside the window of N slots (since the window goes from 1 to N).

The reader is referred to Table I for a detail of the notation. The ratio between the duty cycle d and the per slot update probability ω can be seen as a sort of *load factor*, which we will refer to as such in the results section. We impose the condition $d \leq \omega$, which for a queueing system corresponds to Loynes' instability condition [38], i.e., the arrival rate being higher than the service rate. If the average generation rate of updates is lower than the transmission frequency, the problem simplifies to transmitting updates as they are generated (even if some transmission opportunities are left unused). Practical systems can be considered to be sufficiently far from instability, with d being at least one order of magnitude lower than ω.

We see the scheduling of transmissions in four cases. First, we study a *finite horizon* problem where the duration of the monitoring task is $N < \infty$ slots and M transmissions of updates are scheduled within that window. A scheduling minimizing AoI over the finite horizon N is cast into

$$\min \ \Delta$$
$$\text{s.t.} \sum_{j=1}^{N} \mathcal{T}[j] \leq M - 1 \tag{5}$$

where Δ follows from (4).

For this problem, we investigate both source-agnostic and source-adaptive scheduling. The difference lies in the knowledge of update generations at runtime. The latter can adjust the transmissions depending on whether a fresh update is available or not, whereas the agnostic scheduler defines the transmission instants in advance.

We extend these analyses to an *infinite horizon*, where the objective Δ becomes the expected *long-term average AoI*. Instead of letting N approach infinity in (4), we modify the setup for direct comparison by partitioning time into blocks of N slots and defining the average AoI over the kth block as

$$\Delta_k = \mathbb{E}\left[\frac{1}{N}\sum_{n=kN+1}^{(k+1)N} \delta[n]\right]. \tag{6}$$

Accordingly, we recast the problem as

$$\min \ \Delta = \lim_{K\to\infty} \frac{1}{K+1}\sum_{k=0}^{K} \Delta_k$$
$$\text{s.t.} \sum_{j=n_0+1}^{n_0+N} \mathcal{T}[j] \leq M \quad \text{for all } n_0. \tag{7}$$

In this way, the constraint in (7) imposes an upper limit on the duty cycle $d = M/N$ and explicitly formalizes the time window on which to evaluate it. We remark that any constraint allowing ℓM transmissions to be scheduled every ℓN slots, with $\ell \in \mathbb{N}^+$, would result in the same duty cycle. We pick $\ell = 1$ to have a consistent comparison with the finite horizon.

IV. Finite Time Horizon

A. Source-Agnostic Scheduling

A source-agnostic scheduling optimizes the placement of transmission opportunities within a finite time horizon to minimize the expected AoI. The placement is determined in advance, without the ability to adjust during runtime – for instance, by postponing a transmission if no new information is available.

Since the time horizon is fixed and known in advance, we can compute the average AoI by integrating the area beneath the saw-tooth profile of the instantaneous AoI over the time horizon, as illustrated in Fig. 2. The expected average AoI for a given transmission pattern can be derived using geometric considerations similar to those in [1], [17], and [28]. However, a modification is required from these contributions, as they assume that a fresh update is always sent, thereby resetting the AoI to 0 (or the minimum value) with every successful update. In our setup, fresh information may wait before being transmitted, so the AoI can reset to a higher value.

The area to minimize, corresponding to the average AoI Δ, can be derived as follows. First, instead of the M transmission instants n_1, n_2, ..., n_M, it is convenient to consider the inter-transmission intervals defined as $y_j = n_{j+1} - n_j$ for $j = 0, \dots, M$, where $n_0 = 0$ and $n_{M+1} = N$.

Then, AoI increases linearly (albeit in a discrete fashion) between updates, resulting in the characteristic saw-tooth profile and the inclusion of triangular-like areas in the computation.[2] In fact, this term represents the growth of the AoI even when all the updates are optimal, it is just intrinsic to the behavior of the AoI and resembles what appears in classic problems of dynamic programming [19].

Since the saw-tooth pattern may not reset to 0, the geometric computation involves additional *rectangles* over the y_i periods, whose height is an AoI bias resulting from the lack of updates to a fresh status. Two cases are possible: 1) the most recent update generation during the transmission in slot n_j is not found in slot n_j itself, but still it is inside y_{j-1}, i.e., before slot n_{j-1}; or 2) the entire duration of y_{j-1} is without fresh updates. In the latter case, not only is the whole AoI biased by y_{j-1} (i.e., the height of the rectangle is incrementally increased), but the process is iteratively nested, so that the bias propagates to previous intervals.

We can obtain the average AoI Δ as the sum of the area below the saw-tooth profile, computed through following contributions: (a) the area below the triangles, which is always present even in the case that fresh information is always available; (b) in addition, if a whole inter-update interval y_k is without updates (which happens with probability $(1-\omega)^{y_k}$), a following jth interval, with $j > k$ can contain an additional term equal to the area of a rectangle with basis equal to y_j; and (c) another rectangular area is to be included for the cases where the kth interval is the one getting the last useful update. This results in

$$N\Delta = \underbrace{\sum_{j=0}^{M} \frac{y_j^2 - y_j}{2}}_{\text{(a)}} + \sum_{j=1}^{M} \sum_{k=0}^{j-1} y_j \, \mathcal{U}_{jk}(\omega) \times \Bigg[\underbrace{y_k(1-u)^{y_k}}_{\text{(b)}} + \underbrace{\frac{1-u}{u} \left(1 - y_k(1-\omega)^{y_k - 1} + (y_k - 1)(1-\omega)^{y_k}\right)}_{\text{(c)}} \Bigg] \tag{8}$$

with

$$\mathcal{U}_{jk}(\omega) = \begin{cases} (1-\omega)^{\mathcal{L}}, \text{with } \mathcal{L} = \sum_{\ell=k+1}^{j-1} y_\ell & \text{if } j > k+1 \\ 1 & \text{otherwise}, \end{cases} \tag{9}$$

where (a), (b), and (c) correspond to the three parts described above. Since we consider a discrete time-axis [28], the area of the triangle-like region is $\sum_{k=0}^{y_j} k = (y_j^2 - y_j)/2$. This is the only impact of the discrete time and the other areas are not subject to any rounding. Term (b) can be iterated to any previous interval that goes without updates, and term (c) can be computed as the finite sum of Jonquière's polylogarithm $\mathrm{Li}_{-1}(\omega)$ [39]. Finally, (b) and (c) are connected by a preliminary summation of the intermediate terms connecting j and $k < j$, which is computed in (9). This is equal to 1 if $k = j-1$, else it accounts for the probability that all the intermediate intervals are without updates, so that the bias propagates.

Despite its complex expression, (8) is easy to solve via numerical means. Even the largest scheduling shown in the numerical results took less than 2 s to be computed by means of standard interior points methods when run on M2,2002 chipset with 16GB RAM. Thus, the numerical solution can be used to find a choice of values $y_0, y_1, y_2, \dots, y_M$ that, under the additional condition $\sum_{j=0}^{M} y_j = N$, solve (5).

We remark that the finiteness of the horizon and the initial condition $\delta[0] = 0$ impact on the expression in that the bias does not propagate below y_0. Thus, the first interval has no bias and $y_0 \geq C$, with equality when $\omega \to 1$. This is because if the probability of not generating any update during the first interval of C slots is considerable, the agnostic scheduler will postpone the first transmission. More generally, it will diverge from a periodic pattern, with initial transmissions being postponed, as ω decreases.

B. Source-Aware Scheduling

Now, consider a case where the transmitter is allowed to adjust the transmission instants at run-time, instead of planning them in advance. This makes sense if the transmitter is aware of both current AoI and the value that AoI will be set to, if a transmission is performed. For this reason, we consider a stateful scheduling procedure, fully informed about the current AoI and the age of the freshest available update. A dynamic programming to this problem can be obtained as an MDP; since the horizon is finite, the optimal transmission policy is found via backward induction [19].

The problem can be cast on defining a state, a control vector, and a noise component. In time slot n, with $0 \leq n \leq N$, the state is $x[n] = (\delta[n], w[n], m[n])$, where $\delta[n]$ is the instantaneous AoI in slot n, with initialization as $\delta[0] = 0$. Value $m[n]$ is the number of transmit opportunities still available in n, which is initialized as $m[0] = M-1$, and finally quantity $w[n] = n - z[n]$ is used to represent the time elapsed from the last update generation, and is initialized as $w[0] = 0$. To perform a transmission, $m[n] > 0$ must hold, in which case $m[n]$ is decreased by one and the value of $w[n+1]$ is copied into the AoI.[3] The control $u[n]$, with $0 \leq n \leq N-1$, is a binary choice on whether to transmit based on the state in slot n; $u[n] = 1$ means a transmission happens, whereas 0 means that it does not. Finally, the noise is just the random character in the evolution of $w[n]$, fully captured by probability ω.

Since we allow for the transmission of an update in the same slot it is generated, we obtain that the AoI evolves as

$$\delta[n+1] = \begin{cases} \delta[n] + 1 & \text{if } u[n] = 0 \\ w[n+1] & \text{if } u[n] = 1 \end{cases} \tag{10}$$

and the rest of the state components evolve as

$$w[n+1] = \begin{cases} 0 & \text{with probability } \omega \\ w[n] + 1 & \text{with probability } 1 - \omega \end{cases} \tag{11}$$

[2] In similar studies [17], [33], where time is continuous, these areas form perfect isosceles right triangles. In our computation, the areas are essentially the same, but instead of a straight line, the contours follow a staircase pattern.

[3] Updates can be generated and transmitted in the same slot, so $\delta[n]$ goes as $w[n]$ (same time slot) after a transmission. This is again coming from our conventions, the indices of $w[n]$ can be shifted if another convention is used.

$$m[n+1] = \begin{cases} m[n] & \text{if } m[n] \cdot u[n] = 0 \\ m[n] - 1 & \text{if } m[n] \cdot u[n] > 0 \end{cases} . \quad (12)$$

It immediately follows that the state has the Markov property, so this is well defined as an MDP.

The problem becomes to derive $u[n] = \mu_n(x[n])$, i.e., find the optimal policy $\mu_n(x[n])$ for any state $x[n]$ and $1 \le n \le N$ minimizing the expected value of cost

$$g_n\left(x[n], u[n], \omega\right) = \delta[n] . \quad (13)$$

Bellman's optimality condition [19] can be exploited to find $\mu_{n-1}(x[n-1])$, since if $\mu_0, \ldots, \mu_{N-1}$ describes an optimal policy, then for any value of state $x[n]$ in an intermediate slot n, $0 < n < N-1$, occurring with positive probability, policy $\mu_n, \ldots, \mu_{N-1}$ is the minimizing policy for the residual cost.

The resulting optimal policy, denoted as *source-aware* scheduling, will be time-dependent, e.g., the condition on whether to transmit or not may be different at the beginning or at the end of the time window. Also, in the last time slot, the cost paid will be $\delta[N]$; thus, the control action in the last step imposes to transmit when $m[N-1] > 0$, and conversely the only allowed action when $m[n] = 0$ is not to transmit for every n. So, one can start by defining μ_{N-1} for the last instant and proceed backwards to find the optimal source-aware scheduling for all reachable states in every n.

Induction derives policy μ_n from μ_{n+1} for $0 \le n \le N-2$ as

$$\begin{aligned} &\mu_n\left(x[n], \omega\right) = \mathbb{1}\left[a(x[n]) > b(x[n])\right] \quad \text{where} \\ &\quad a(\delta, w, m) = \omega R_n(w, 0, m-1) + (1-\omega) R_n(w, w+1, m-1) \\ &\quad b(\delta, w, m) = \omega R_n(\delta+1, 0, m) + (1-\omega) R_n(\delta+1, w+1, m) \\ &\quad R_n(x[n]) = \sum_{i=n}^{N} g_i\left(x[i], \mu_i(x[i]), \omega\right) \end{aligned} \quad (14)$$

with $\mathbb{1}[\cdot]$ being an indicator function equal to 1 if the condition is true, 0 otherwise. The optimal policy in slot n minimizes the expected AoI accounting for its evolution in $n+1, \ldots, N$, assuming future decisions are optimal ($\mu_{n+1}, \ldots, \mu_N$ are determined), averaging over update generations. The essence of the decision in (14) is to compare (a) transmitting, so that $w[n]$ is copied into $\delta[n]$ and $m[n]$ is decreased by 1, versus (b) not transmitting, i.e., m is unchanged and AoI increases by 1.

The optimal control policy in principle requires the definition of a matrix of $O(N^3 M)$ binary elements, since it must specify the control action (transmit/not transmit) for every n, $\delta[n]$, $w[n]$, and $m[n]$. A better implementation can be obtained by exploiting the threshold property of an optimal scheduler, which is immediate and posits that, $\mu_n(\delta, w, m) \le \mu_n(\delta, w', m)$ for every $w' < w$. In other words, if a transmission is performed when AoI is δ and the update would bring it to w, it is also performed when the update is fresher. As a result, an equivalent but more efficient implementation, i.e., with complexity $O(N^2 M)$, of $\mu_n(\delta, w, m)$ may be given as $\psi_n(\delta, m)$ being a non-negative integer such that $\mu_n(\delta, w, m) = \mathbb{1}[w < \psi_n(\delta, m)]$.

We compare the schedules in Fig. 3, showing the average transmission instants of the source-aware policy, compared with those of the agnostic scheduler, for different values of ω in the same setup of four transmissions over 1000 slots. Since a source-aware schedule is variable over different realizations, its average transmission instants are shown with statistical dispersion (the vertical bars are the standard deviation). If $\omega \gg d$, both approaches lead to a periodic schedule. As ω decreases, the transmissions are postponed. This is more pronounced in the source-agnostic scheduling than (on average) in the source-aware scheduling. However, the latter sometimes postpones the updates more significantly, only this happens relatively rarely, and with higher statistical variability, for low ω.

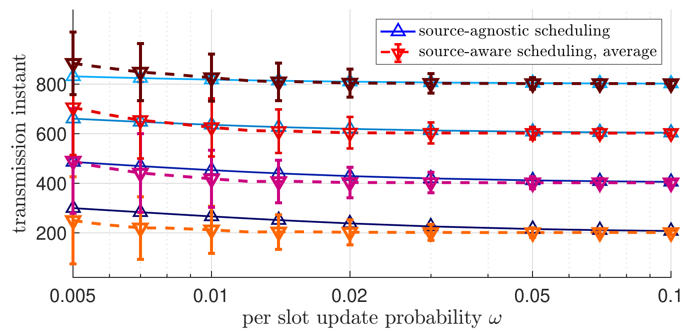


Fig. 3. Scheduling of four transmissions over a finite time horizon of 1000 slots, versus per slot update probability ω. Optimal agnostic scheduling in shades of blue, average of source-aware scheduling in shades of red.

V. Infinite Time Horizon

A. Source-Agnostic Scheduling

A source-agnostic scheduling performed over an infinite time horizon gives a transmission every C slots, with duty cycle equal to $1/C$. Thus, the AoI for an exogenous generation with probability ω is obtained by considering cycles of C slots. Over each cycle, the average AoI contains the term $(C^2 - C)/2$ due to sending an update every C slot. This term is present even under generate at will. In addition, the average AoI has an initial bias due to the instantaneous AoI not being necessarily 0 when updating. Such a bias can be found to be equal to $C(1-\omega)/\omega$, since the initial value of the AoI is i with probability $\omega(1-\omega)^i$, and the bias is kept for all C slots of a cycle. Thus, the average AoI can be obtained by dividing the integral by the duration of a cycle. This results in

$$\Delta = \frac{(C^2 - C)/2 + C(1-\omega)/\omega}{C} = \frac{C}{2} - \frac{1}{2} + \frac{1-\omega}{\omega} . \quad (15)$$

B. Source-Aware Scheduling

While a source-aware scheduling with infinite horizon is the most common setup in the related literature [11], [12], [13], [14], [27], [34], its implementation for our problem is not easy in relation to computational issues. The standard approach to compute the optimal policy for an MDP consists of value iteration [40], which generalizes the backward induction seen in the previous problem. This requires to include a discount factor $\gamma \in (0, 1)$ in the long term cost function of (14) as

$$R(x[0]) = \sum_{i=0}^{\infty} \gamma^i g_i\left(x[i], \mu_i(x[i]), \omega\right) \quad (16)$$

that allows the (discounted) sum to converge and estimate for the expected future rewards even in an infinite horizon setup.

However, there are several challenges for the problem at hand. First, the cost function in (13) does not depend on n. Hence, value iteration is efficient in finding a time-independent scheduling policy [19] but a desirable policy would be time dependent. Also, the use of an exponentially discounted cost and value iteration is appropriate for a stationary system. The system at hand is not stationary since the available transmission opportunities vary according a pattern of M every N slots.

Moreover, even though the introduction of a discount factor is not an issue in many engineering setups, for the problem at hand it may result in a noticeable approximation, as it implies that future costs k slots in the future are discounted by a factor γ^k. Thus, γ must approach 1 as k can be significantly large in numerical terms (if the duty cycle d is less than 1%, we need $k > 10^3$). All of this may result in either a coarse approximation or a formidable computational complexity.

Thus, we derive two simplified versions of the source-aware scheduler. First, a *time independent* procedure where the transitions in (12) are changed to allow for a stationary system and derive a time-independent policy. Then, we propose a *cyclic control* schedule, where the finite horizon control is repeated thanks to the Markov property of the system.

Time independent infinite horizon schedule. This applies to the system at hand a stationary control that is conceived for a similar, but different, system where, instead of a fixed duty cycle constraint, we consider a long-term average of transmissions. In other words, even though the system is originally time dependent, as there is an N-step memory that prevents to perform more than M transmissions in the last N slots, we ignore this dependency and just consider the availability of transmissions to increase at each slot with independent and identically distributed probability equal to M/N. This changes the system evolution as an MDP characterized by

$$m[n] = \begin{cases} m[n{-}1] + q & \text{if } m[n{-}1] \cdot u[n] = 0 \\ m[n{-}1] - 1 + q & \text{if } m[n{-}1] \cdot u[n] > 0 \end{cases}$$

$$\text{with } q = \begin{cases} 1 & \text{with probability } M/N \\ 0 & \text{with probability } 1 - M/N \end{cases} , \tag{17}$$

and allows for a stationary control $\mu^{\mathrm{TI}}(x[n])$ that does not depend on n, which is however only an approximation for the original time-dependent system. The derivation of $\mu^{\mathrm{TI}}(x[n])$ is promptly obtained, e.g., through value iteration [40].

Cyclic control infinite horizon schedule. A better approximation can be found by exploiting a finite horizon solution, and applying the principle of cyclic control [41]. The optimal source-aware scheduling found in the previous section would work reasonably well for an infinite horizon as well, because of the Markov nature of the problem and the backward induction procedure used to derive it, as per (14). The only blunders of a finite horizon policy are for μ_{N-1} and μ_{N-2}. Recall that in the last slot, the policy forced a transmission if $m[n-1] > 0$, since it makes no sense to save transmissions for a nonexisting future. However, in an infinite setup, N is not the end of the horizon but just the last slot of a cycle. Also, the policy in time slot $N{-}2$ accounts for that since μ is recursively computed in steps of 1. Due to the Markov property, the system memory vanishes at step $N{-}3$, where the policy is then applicable to an infinite horizon.

We can remove the last and second-to-last values of the optimal policy, i.e., μ_{N-2} and μ_{N-1} and make it cyclical. We set the number of allowed transmissions as M (opposed to $M{-}1$ as in the finite horizon case) since there is no AoI reset at slot 0 and therefore no initialization $\delta[0] = 0$. This means that we derive the optimal finite-horizon policy $\mu_n(x[n])$ for N time instants (i.e., $0 \leq n \leq N$), M transmission opportunities, and generation rate of updates equal to ω.

The cyclic control $\mu_n^{\mathrm{CC}}(x[n])$ for an infinite horizon is $\mu_n^{\mathrm{CC}}(x[n]) = \mu_k(x[k])$ where:

$$k = \begin{cases} 0 & \text{if } \mathrm{mod}(n, N) \leq 2 \\ \mathrm{mod}(n, N) - 2 & \text{otherwise} \end{cases} . \tag{18}$$

C. Theoretical Upper Bounds

Finally, we introduce some theoretical performance bounds that will be further validated in the numerical evaluations. In the worst case there are constant multiplicative limits of 2 and 3 for source-aware and agnostic scheduling, respectively, and these bounds are much tighter in practical cases of finite horizon and generation rate of updates that is above the minimum duty cycle.

Theorem 1: For exogenous arrivals, the average AoI Δ of an optimal source-aware scheduling satisfies $\Delta \leq 2\Delta_{\mathrm{gw}}$, where Δ_{gw} is the average AoI under generate at will.

Proof: Under generate at will, the AoI always resets to 0 when a transmission is performed. If the number of transmissions is limited by duty cycle $d = M/N$, the optimal schedule is the one that transmits every $C = 1/d$ slots, thereby achieving average AoI $\Delta_{\mathrm{gw}} = (C-1)/2$.

Now, consider the case where fresh updates are only generated with probability ω at each slot. This implies that transmissions cannot be performed arbitrarily, but have to follow the generation process, which can result in uneven scheduling of transmissions. Due to the stability condition $d \geq \omega$, the worst case is that $\omega = C^{-1}$, i.e., the lowest possible generation rate. However, since the long-term average number of allowed transmissions is equal to the generated updates, the optimal source-aware scheduling is to transmit a fresh update whenever available. Thus, the intertransmission time is geometrically distributed with parameter $1-\omega$, which leads to the known result that the average AoI is $\Delta = \omega^{-1} - 1$. Since the worst case of exogenous transmissions is $\omega = C^{-1}$, we get that $\Delta \leq C - 1$, which implies that $\Delta/\Delta_{\mathrm{gw}} \leq 2$. ■

Theorem 2: Under exogenous arrivals, the average AoI Δ of an optimal agnostic scheduling satisfies $\Delta_{\mathrm{gw}} \leq \Delta \leq 3\Delta_{\mathrm{gw}}$.

Proof: The result follows from the finding of previous Theorem 1 that $\Delta_{\mathrm{gw}} = (C-1)/2$. Under $\omega \geq C^{-1}$, we see that (15) implies $\Delta \leq 3(C-1)/2$ for an agnostic scheduler, with equality holding in the worst case of one generation every C slots on average. Thus, $\Delta/\Delta_{\mathrm{gw}} \leq 3$. ■

Also note the following about these theorems.

Remark 1: The upper bounds of Theorems 1 and 2 hold for an infinite horizon. They apply to the finite horizon as well, but they are looser. Since the instantaneous AoI starts from zero at the beginning, an initial transient phase is observed where the average AoI (over a finite interval) is lower.

And finally, we point out that these bounds only hold for very low ω (close to the minimum allowed value), but as we will see through numerical investigations, the AoI surge is steep around

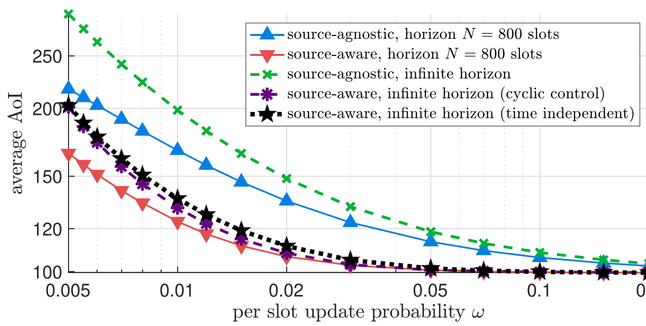


Fig. 4. Average AoI under finite/infinite and source-agnostic/aware scheduling versus per slot update probability ω, with duty cycle 0.005.

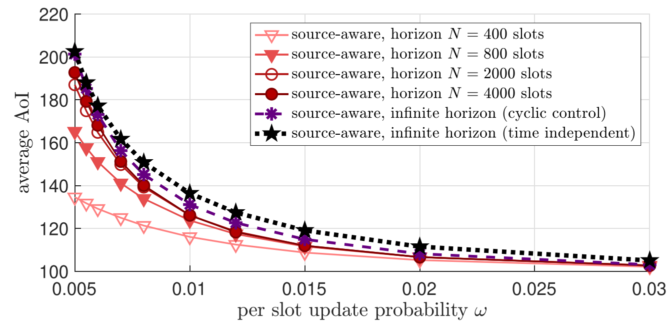


Fig. 6. Average AoI under source-aware scheduling for different time horizons versus per slot update probability ω, with duty cycle 0.005.

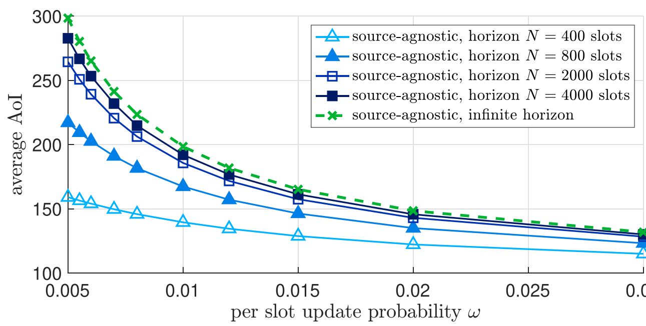


Fig. 5. Average AoI under source-agnostic scheduling for different time horizons versus per slot update probability ω, with duty cycle 0.005.

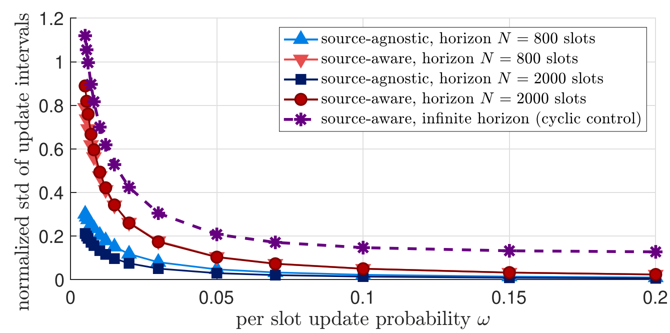


Fig. 7. Normalized standard deviation of inter-update intervals for different policies versus per slot update probability ω, duty cycle 0.005.

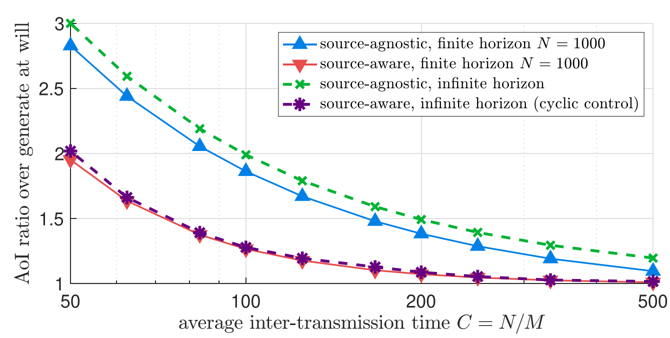


Fig. 8. AoI amplification over generate at will vs average inter-transmission time N/M, per slot update probability $\omega = 0.02$.

$\omega \to C^{-1}$, which implies that in the case of update generation that is neither persistent nor too sporadic, the average AoI is similar to the generate at will case.

VI. Numerical Results

We present numerical evaluations to assess exogenous update generation. The average AoI is taken as the objective and we quantify its surges when the idealized assumptions of the literature do not hold, which is of interest for industrial setups.

Fig. 4 compares the four approaches of finite/infinite horizon and source agnostic/aware scheduling, with duty cycle $d = M/N = 0.005$. For infinite horizon and source-aware scheduling, two implementations are given, i.e., the time independent and the cyclic control policies, showing how they agree with each other, yet the former slightly overestimates the average AoI. For larger ω, fresh updates are more frequent, the average AoI decreases, and all strategies tend to similar performance.

This is expanded in Figs. 5 and 6 to highlight the impact of the growth in N toward an infinite horizon, for source-agnostic and source-aware policies, respectively. The y-axis is in linear scale and the range of the x-axis is smaller than Fig. 4 to appreciate the differences between the curves, since for larger ω, all the policies basically perform the same.

Fig. 7 shows instead the normalized standard deviation of the scheduling policies, again for a duty cycle $d = 0.005$. The agnostic scheduling over an infinite horizon performs transmissions at regular intervals, hence its standard deviation is zero. Variations are larger for a source-aware scheduling, especially it explodes as ω tends to d. Variability in the agnostic schedules just relates to the difference between fixed times, whereas for the source-aware policies is computed over different realizations (the average transmission instants of a source-aware scheduling are instead *less* variable, see Fig. 3).

Fig. 8 considers a fixed ω and a variable average intertransmission time C. The increase in AoI over the case with status updates generated at will is shown to be limited as long as d is one order of magnitude lower than ω, which can be taken as a de facto persistent generation of updates. As d increases, updates become scarcer, and the AoI ratio over the generate at will increases to 2 in the source-aware case, and 3 in the agnostic case, when the duty cycle approaches ω. These increases are slightly lower if the horizon is finite.

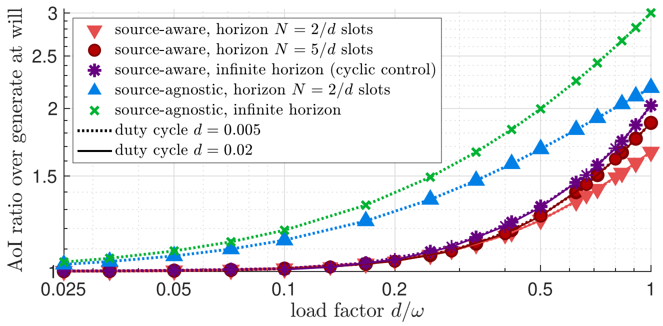


Fig. 9. AoI amplification over generate at will versus load d/ω.

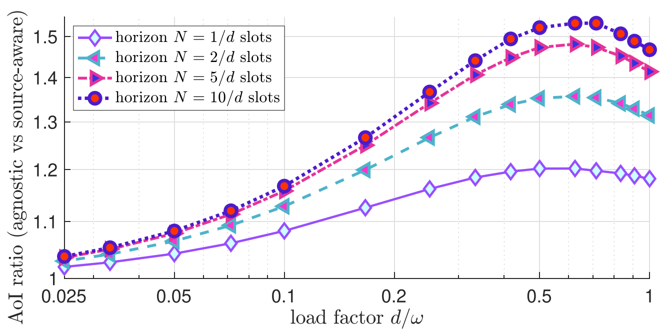


Fig. 10. AoI increase of agnostic scheduling vs load d/ω.

Fig. 9 shows the increase in the average AoI depending on the load factor (the ratio between the duty cycle d and the per slot update probability ω). The most extreme increase factor of AoI due to exogenous update generation are of 3 and 2 times, for source-agnostic and source-aware scheduling, respectively. However, this is true only for infinite horizon and load factor very close to 1, otherwise the increase is much more contained. As argued in Section III, a reasonable condition can be that the load factor is below 0.1, that is, while update generation is still sporadic, it is performed with one order of magnitude more than the transmission. In this case, as visible from Fig. 9, the increase in AoI is almost negligible for a source-aware scheduling, and also less than 20% an agnostic one, and the values are even lower in the case of a finite horizon.

Fig. 10 shows the ratio of average AoI values between source-agnostic and source-aware schedulers, in a finite horizon. The two approaches perform similarly in a wide range of load factor values, especially if the horizon is limited. This might affect scheduling for task-oriented IIoT, whenever the generation rate of updates is not excessively low without being persistent; if so, an agnostic policy may be convenient and a source-aware implementation, if available, may be preferable only when the increase in the average AoI is significant.

These considerations must be weighed upon the specific scenario. Whether a given increase in AoI is acceptable, or a source-aware scheduling is convenient over the agnostic version, ultimately depends on the application. Still, we do not really need the update generation to be persistent to approach the performance of the generate at will case.

VII. Conclusion

We analyzed the impact of exogenous update generation over AoI in the IIoT. We investigated four different conditions of finite/infinite horizon and source-aware versus agnostic scheduling. We derived a quantitative assessment of the increase in AoI due to rare generation of updates. One heavy limitation of IIoT is scalability, especially since some systems especially since some IIoT systems might involve a vast number of sensors and actuators [18]. In such a scenario, coordination of multiple agents is difficult to achieve and distributed state-agnostic solutions are preferrable, even though our results show that they can be underperforming. The same holds when the network topology is variable and/or random delay effects may be present [17]. These problems also typically arise in larger networks and may cause the preference of the network manager to lean towards simpler agnostic policies.

Another important factor to keep into consideration is that of power consumption, which is another limiting factor in IIoT systems [20], [34]. A source-aware scheduling policy requires a more frequent monitoring of the process at the sensor's side, to adjust the transmission pattern accordingly, which may lead to a higher energy expenditure.

However, especially for critical alerts or emergency data, responsiveness of the system is key, and our results show that a source-aware scheduler is advisable, especially in light of its close to optimal performance even under sporadic data generation. Extensions involving an exogenous generation of updates that depends on specific aspects, either in the semantic realm such as alert generation/retransmissions [7], [11], or energy harvesting [33], [35] can be envisioned in future work.

Finally, another important aspect, transversal to all the considered policies, is that of incorporating considerations about correctness of information and/or security within the problem of AoI minimization, so as to ensure data integrity and confidentiality [30], [42]. In this sense, neither agnostic nor source-aware schedulers are inherently better, since the former operate in systems with little supervision, while the latter involve a heavier data exchange, which may be compromised more easily. At any rate, thwarting malicious attacks towards data freshness is an important aspect to be considered in future research.

References

[1] S. K. Kaul, R. D. Yates, and M. Gruteser, "Real-time status: How often should one update?," in *Proc. IEEE INFOCOM*, 2012, pp. 2731–2735.

[2] C. Kam, S. Kompella, G. D. Nguyen, J. E. Wieselthier, and A. Ephremides, "On the age of information with packet deadlines," *IEEE Trans. Inf. Theory*, vol. 64, no. 9, pp. 6419–6428, Sep. 2018.

[3] B. Zhou and W. Saad, "Performance analysis of age of information in ultra-dense Internet of Things (IoT) systems with noisy channels," *IEEE Trans. Wireless Commun.*, vol. 21, no. 5, pp. 3493–3507, May 2022.

[4] Y. Sun, E. Uysal-Biyikoglu, R. D. Yates, C. E. Koksal, and N. B. Shroff, "Update or wait: How to keep your data fresh," *IEEE Trans. Inf. Theory*, vol. 63, no. 11, pp. 7492–7508, Nov. 2017.

[5] M. Xie, Q. Wang, J. Gong, and X. Ma, "Age and energy analysis for LDPC coded status update with and without ARQ," *IEEE Internet Things J.*, vol. 7, no. 10, pp. 10388–10400, Oct. 2020.

[6] A. E. Kalør and P. Popovski, "Minimizing the age of information from sensors with common observations," *IEEE Wireless Commun. Lett.*, vol. 8, no. 5, pp. 1390–1393, Oct. 2019.

[7] F. Chiariotti et al., "Query age of information: Freshness in pull-based communication," *IEEE Trans. Commun.*, vol. 70, no. 3, pp. 1606–1622, Mar. 2022.

[8] X. Xie, H. Wang, and X. Liu, "Scheduling for minimizing the age of information in multi-sensor multi-server industrial IoT systems," *IEEE Trans. Ind. Informat.*, vol. 20, no. 1, pp. 573–582, Jan. 2024.

[9] H. Wang, X. Xie, and J. Yang, "Optimizing average age of information in industrial IoT systems under delay constraint," *IEEE Trans. Ind. Informat.*, vol. 19, no. 10, pp. 10244–10253, Oct. 2023.

[10] O. T. Yavascan, M. Ahmetoglu, E. Uysal, U. Gurbuz, O. Gencay, and A. Balci, "DCCTA: Age of information in slotted ALOHA under duty cycle constraints," in *Proc. IEEE Int. Conf. Comput., Netw. Commun.*, 2024, pp. 371–375.

[11] E. T. Ceran, D. Gündüz, and A. György, "Average age of information with hybrid ARQ under a resource constraint," *IEEE Trans. Wireless Commun.*, vol. 18, no. 3, pp. 1900–1913, Mar. 2019.

[12] C. Li, S. Li, Y. Chen, Y. T. Hou, and W. Lou, "AoI scheduling with maximum thresholds," in *Proc. IEEE INFOCOM 2020 - IEEE Conf. Computer Commun.*, 2020, pp. 436–445.

[13] Q. Liu, C. Li, Y. T. Hou, W. Lou, and S. Kompella, "AION: A bandwidth optimized scheduler with AoI guarantee," in *Proc. 2021 IEEE Conf. Computer Commun.*, 2021, pp. 1–10.

[14] M. Hatami, M. Leinonen, and M. Codreanu, "AoI minimization in status update control with energy harvesting sensors," *IEEE Trans. Commun.*, vol. 69, no. 12, pp. 8335–8351, Dec. 2021.

[15] T.-W. Kuo, "Minimum age of information TDMA scheduling: Approximation algorithms and hardness results," *IEEE Trans. Inf. Theory*, vol. 66, no. 12, pp. 7652–7671, Dec. 2021.

[16] F. Liu, W. S. Wong, Y.-H. Lo, Y. Zhang, C. S. Chen, and G. Xing, "Age of information for periodic status updates under sequence based scheduling," *IEEE Trans. Commun.*, vol. 71, no. 10, pp. 5963–5978, Oct. 2023.

[17] L. Badia, A. Zancanaro, G. Cisotto, and A. Munari, "Status update scheduling in remote sensing under variable activation and propagation delays," *Ad Hoc Netw.*, vol. 163, Oct. 2024, Art. no. 103583.

[18] L. Corneo, C. Rohner, and P. Gunningberg, "Age of information-aware scheduling for timely and scalable Internet of Things applications," in *Proc. IEEE INFOCOM IEEE Conf. Computer Commun.*, 2019, pp. 2476–2484.

[19] D. Bertsekas, *Dynamic Programming and Optimal Control: Volume I*. Nashua, NH, USA: Athena Scientific, 2012.

[20] P. Hegde, L. Badia, and A. Munari, "Age of information for remote sensing with uncoordinated finite-horizon access," *ICT Exp.*, vol. 10, no. 4, pp. 786–791, Aug. 2024.

[21] C. Ari, M. K. C. Shisher, E. Uysal, and Y. Sun, "Goal-oriented communications for remote inference under two-way delay with memory," in *Proc. IEEE Int. Symp. Inf. Theory*, 2024, pp. 1179–1184.

[22] T. M. Getu, G. Kaddoum, and M. Bennis, "Making sense of meaning: A survey on metrics for semantic and goal-oriented communication," *IEEE Access*, vol. 11, pp. 45456–45492, 2023.

[23] J. Holm, F. Chiariotti, A. E. Kalør, B. Soret, T. B. Pedersen, and P. Popovski, "Goal-oriented scheduling in sensor networks with application timing awareness," *IEEE Trans. Commun.*, vol. 71, no. 8, pp. 4513–4527, Aug. 2023.

[24] S. Kaul, M. Gruteser, V. Rai, and J. Kenney, "Minimizing age of information in vehicular networks," in *Proc. 8th Annu. IEEE Commun. Soc. Conf. Sensor, Mesh Ad Hoc Commun. Netw.*, 2011, pp. 350–358.

[25] J. P. Champati, H. Al-Zubaidy, and J. Gross, "On the distribution of AoI for the GI/GI/1/1 and GI/GI/1/2* systems: Exact expressions and bounds," in *Proc. IEEE INFOCOM IEEE Conf. Computer Commun.*, 2019, pp. 37–45.

[26] O. T. Yavascan and E. Uysal, "Analysis of slotted ALOHA with an age threshold," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 5, pp. 1456–1470, May 2021.

[27] E. Fountoulakis, N. Pappas, M. Codreanu, and A. Ephremides, "Optimal sampling cost in wireless networks with age of information constraints," in *Proc. IEEE INFOCOM IEEE Conf. Computer Commun. Workshops*, 2020, pp. 918–923.

[28] A. Javani, M. Zorgui, and Z. Wang, "On the age of information in erasure channels with feedback," in *Proc. IEEE Int. Conf. Commun.*, 2020, pp. 1–10.

[29] L. Crosara and L. Badia, "A stochastic model for age-of-information efficiency in ARQ systems with energy harvesting," in *Proc. Eur. Wireless 2021; 26th Eur. Wireless Conf.*, 2021, pp. 1–6.

[30] A. Asheralieva and D. Niyato, "Optimizing age of information and security of the next-generation internet of everything systems," *IEEE Internet Things J.*, vol. 9, no. 20, pp. 20331–20351, Oct. 2022.

[31] N. Akar and E. O. Gamgam, "Distribution of age of information in status update systems with heterogeneous information sources: An absorbing markov chain-based approach," *IEEE Commun. Lett.*, vol. 27, no. 8, pp. 2024–2028, Aug. 2023.

[32] K.-Y. Lin, Y.-C. Huang, and Y.-P. Hsu, "Scheduling for periodic multi-source systems with peak-age violation guarantees," *IEEE Trans. Commun.*, vol. 71, no. 12, pp. 7102–7116, Dec. 2023.

[33] R. D. Yates, "Lazy is timely: Status updates by an energy harvesting source," in *Proc. IEEE Int. Symp. Inf. Theory*, 2015, pp. 3008–3012.

[34] S. Feng and J. Yang, "Age of information minimization for an energy harvesting source with updating erasures: Without and with feedback," *IEEE Trans. Commun.*, vol. 69, no. 8, pp. 5091–5105, Aug. 2021.

[35] E. Gindullina, L. Badia, and D. Gündüz, "Age-of-information with information source diversity in an energy harvesting system," *IEEE Trans. Green Commun. Netw.*, vol. 5, no. 3, pp. 1529–1540, Sep. 2021.

[36] H. Tang, Y. Chen, J. Wang, P. Yang, and L. Tassiulas, "Age optimal sampling under unknown delay statistics," *IEEE Trans. Inf. Theory*, vol. 69, no. 2, pp. 1295–1314, Feb. 2023.

[37] K. Saurav and R. Vaze, "Minimizing the sum of age of information and transmission cost under stochastic arrival model," in *Proc. IEEE INFOCOM IEEE Conf. Computer Commun.*, 2021, pp. 1–10.

[38] R. M. Loynes, "The stability of a queue with non-independent inter-arrival and service times," *Math. Proc. Cambridge Phil. Soc.*, vol. 58, no. 3, 1962, pp. 497–520.

[39] W. Magnus, F. Oberhettinger, and R. P. Soni, *Formulas and Theorems for the Special Functions of Mathematical Physics*. Berlin, Germany: Springer Verlag, 1966.

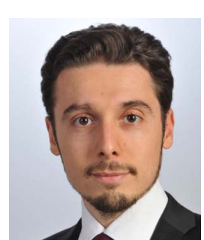
[40] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Hoboken, NJ, USA: John Wiley & Sons, Inc., 2014.

[41] S. D. Flåm and R. J. B. Wets, "Existence results and finite horizon approximates for infinite horizon optimization problems," *Econometrica*, pp. 1187–1209, 1987.

[42] V. Bonagura, S. Panzieri, F. Pascucci, and L. Badia, "Strategic control against an intruder for timely and accurate updates to a reactive receiver," in *Proc. Eur. Control Conf.*, 2024, pp. 335–340.

Leonardo Badia (Senior Member, IEEE) received the Laurea degree (Hons.) in electrical engineering and the Ph.D. degree in information engineering from the University of Ferrara, Ferrara, Italy, in 2000 and 2004, respectively.

From 2002 to 2003, he was Visiting Scholar with the Radio System Technology Labs (currently, Wireless@KTH), Royal Institute of Technology, Stockholm, Sweden. After having been with the Engineering Department, University of Ferrara, he joined in 2006 the IMT Institute for Advanced Studies, Lucca, Italy. In 2011, he moved to the University of Padua, Italy, where he is currently Associate Professor. He has authored or coauthored more than 250 research papers. His scientific interests include protocol design for multihop networks, cross-layer optimization of wireless communication, transmission protocol modeling, and applications of game theory to radio resource management.



Andrea Munari (Senior Member, IEEE) received the M.Sc. and Ph.D. degrees in telecommunications engineering from the University of Padova, Padua, Italy, in 2006 and 2010, respectively.

From 2007 to 2010, he was with IBM Research in Zurich, Switzerland. In 2011, he joined the Corporation Research and Development Division, Qualcomm Inc., San Diego, CA, US. He is currently with the Institute of Communications and Navigation, German Aerospace Center (DLR), Wessling, Germany. His main research interests include, among others, the design and modeling of medium access techniques, with special attention to Internet-of-Things applications.